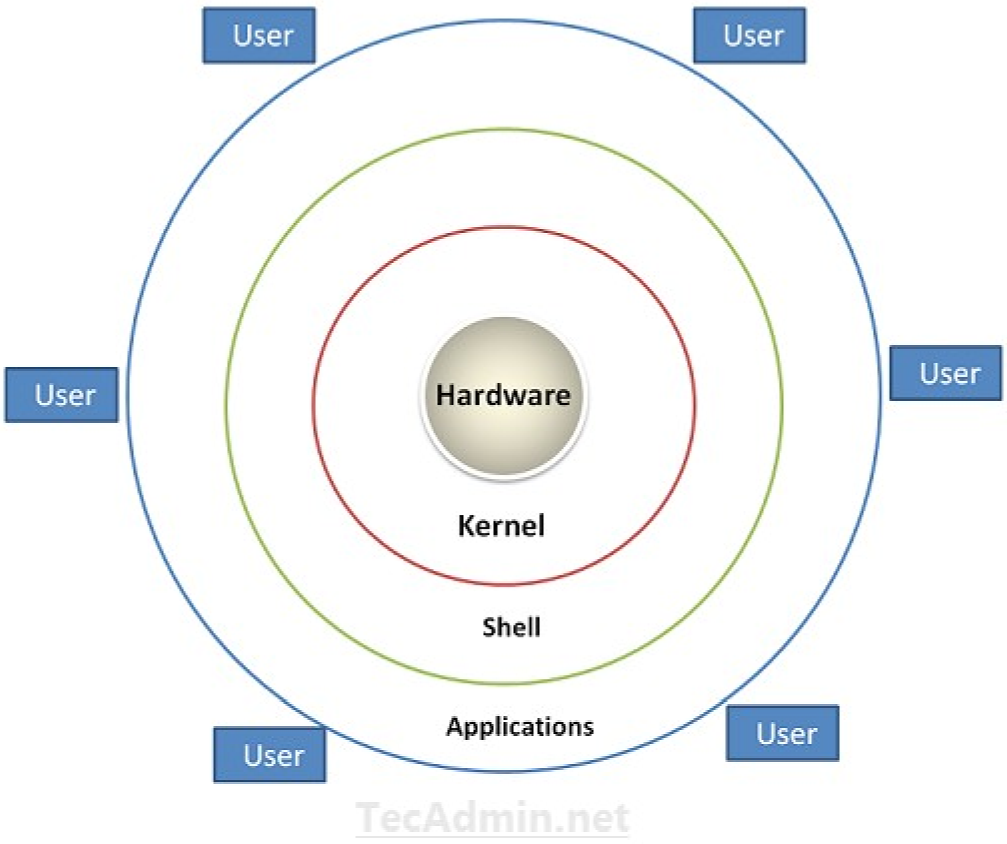


**Linux System Architecture**

# Linux

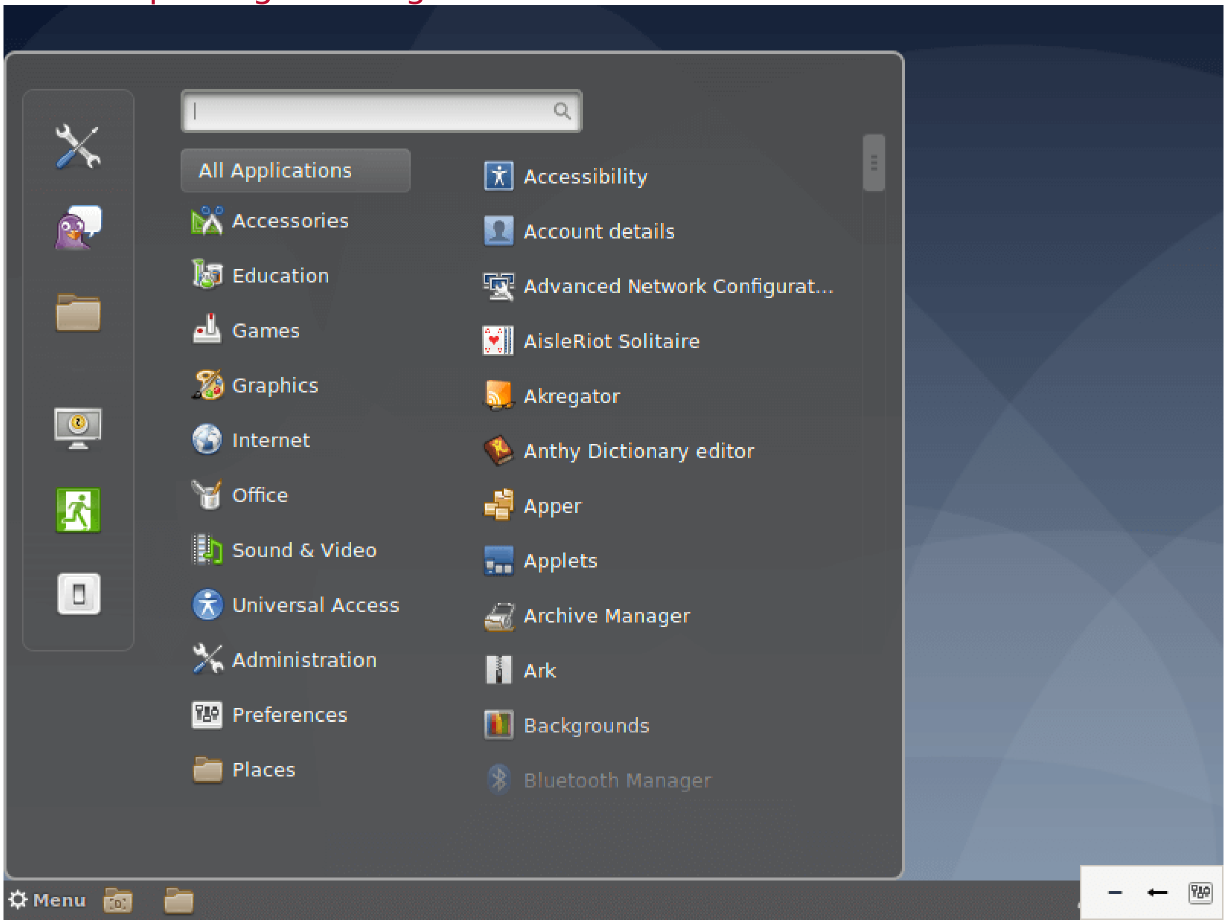
## History : -

1960 Development age
Minix – **Ken Thompson and Dennis Ritchie -** B Language
Unix - **Dennis Ritchie -** C Language
Linus Benedict Torvalds – 1991 -> 1993 -> Linux
95% server of world are based on Linux / Unix
10,000 organizations work for development of Linux.

# Distribution of Linux:-

## 1. Debian

Debian is renowned for being a mother to popular Linux distributions such as **Deepin**, **Ubuntu,** and **Mint** which have provided solid performance, stability, and unparalleled user experience. The latest stable release is **Debian 10.5**, an update of **Debian 10** colloquially known as **Debian Buster**.
Note that **Debian 10.5** does not constitute a new version of **Debian Buster** and is only an update of **Buster** with the latest updates and added software applications. Also included are security fixes that address pre-existing security issues. If you have your **Buster** system, there's no need to discard it. Simply perform a system upgrade using the APT package manager.



Debian Linux

The **Debian** project provides over **59,000** software packages and supports a wide range of PCs with each release encompassing a broader array of system architectures. It strives to strike a balance between cutting edge technology and stability. Debian provides 3 salient development branches: **Stable**, **Testing,** and **Unstable**.

The stable version, as the name suggests is rock-solid, enjoys full security support but unfortunately, does not ship with the very latest software applications. Nevertheless, It is ideal for production servers owing to its stability and reliability and also makes the cut for relatively conservative desktop users who don’t really mind having the very latest software packages. Debian Stable is what you would usually install on your system.

**Debian Testing** is a rolling release and provides the latest software versions that are yet to be accepted into the stable release. It is a development phase of the next stable Debian release. It’s usually fraught with instability issues and might easily break. Also, it doesn’t get its security patches in a timely fashion. The latest **Debian Testing** release is **Bullseye**.
The unstable distro is the active development phase of Debian. It is an experimental distro and acts as a perfect platform for developers who are actively making contributions to the code until it transitions to the ‘Testing’ stage.

Overall, Debian is used by millions of users owing to its package-rich repository and the stability it provides especially in production environments.

## 2. Gentoo

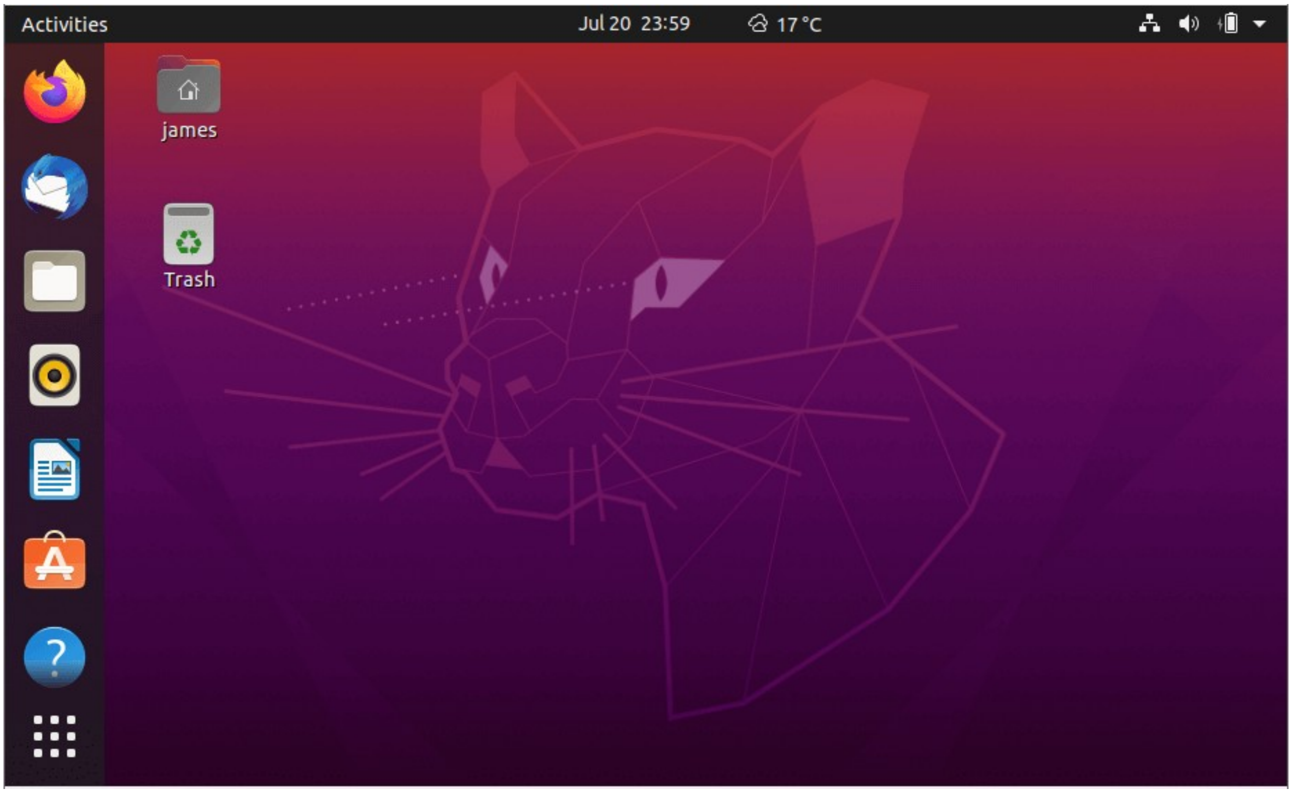
Gentoo is a distro built for professional use and experts who take into consideration what packages they are working with from the word go. This category includes developers, system & network administrators. As such, it’s not ideal for beginners in Linux. **Gentoo** comes recommended for those who want to have a deeper understanding of the ins and outs of the Linux operating system.

Gentoo Linux

**Gentoo** ships with a package management system known as **portage** which is also native to other distros such as Sabayon, and Calculate Linux which is based on **Gentoo** and backward-compatible with it. It is Python-based and based on the ports' collections concept. Port collections are sets of patches and makefiles provided for by BSD-based distros such as OpenBSD and NetBSD.

## 3. Ubuntu

Created and maintained by **Canonical**, **Ubuntu** is one of the most popular Linux distros enjoyed across the globe by beginners, intermediate users, and professionals alike. **Ubuntu** was specifically designed for beginners in Linux or those transitioning from mac and Windows.



Ubuntu Linux

By default, **Ubuntu** ships with **GNOME** desktop environment with every day out-of-the-box applications such as Firefox, LibreOffice, and image editing applications such as **GIMP**, music players, and video players such as **Audacious** and **Rhythmbox**.

The latest version is Ubuntu 20.04 LTS codenamed **Focal Fossa**. It provides numerous improvements and new features such as the new **Yaru** theme, new look and polished icons, great support for Snap packages, and the fractional scaling functionality that provides support for high-resolution displays.

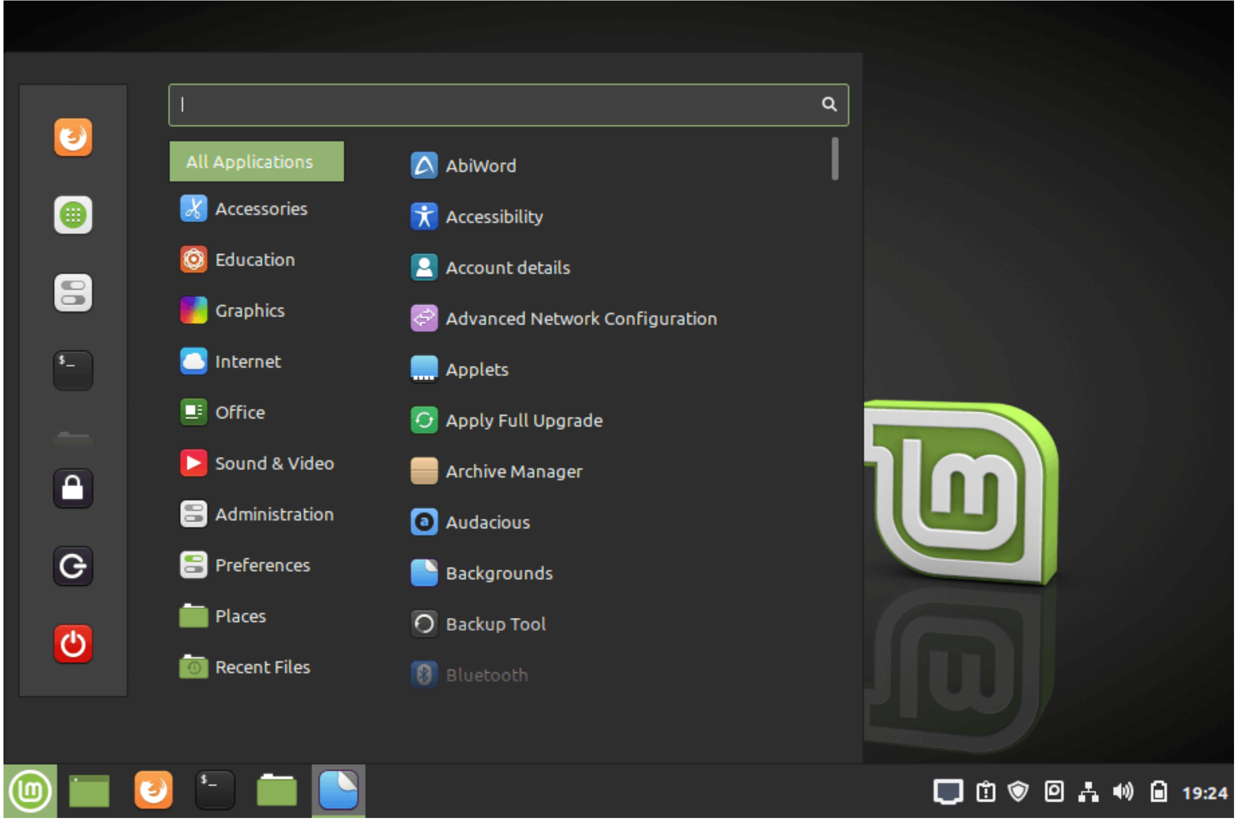
Ubuntu forms the basis of several other Linux distributions. Some of the distributions based on Ubuntu 20.04 include **Lubuntu 20.04 LTS**, **Kubuntu 20.04,** and **Linux Mint 20.04 LTS (Ulyana)**.

Due to its user-friendliness and elegant UI, **Ubuntu** is ideal for desktop users and newcomers who are trying to wrap their head around Linux. They can readily get started with default Apps as stated earlier on as they work their way towards getting a better understanding of Linux.

It’s worth mentioning **Ubuntu Studio** which is geared towards multimedia production. It targets creatives who are looking to make a career in graphics, photography, audio, and video production.

## 4. Linux Mint

Linux Mint is a hugely popular community-driven Linux distro based on **Ubuntu**. It has transcended time to provide one of the most elegant, and user-friendly distributions loved by desktop users and professionals alike. Despite the controversy surrounding the latest release – **Mint 20** – dropping **snap** support by default, Mint remains a stable, powerful and outstanding Linux distribution.



Linux Mint Desktop

To enable **snap** support, simply run the commands:

```
$ sudorm /etc/apt/preferences.d/nosnap.pref

$ sudo apt update

$ sudo apt install snapd
```

Based on **Ubuntu 20.04 LTS**, **Mint 20** is available in 3 desktop editions – **Cinnamon**, **XFCE** and **MATE** editions. Mint has dropped support for 32-bit versions and is only available in 64-bit. Under the hood, **Linux Mint 20** rides on Linux kernel 5.4 with new enhancements such as improved support for AMD Navi 12, Intel Tiger Lake CPU and NVIDIA GPU. Additionally, the general UI has received a revamp with polished icons, new themes, high-resolution background images and a retouched taskbar.

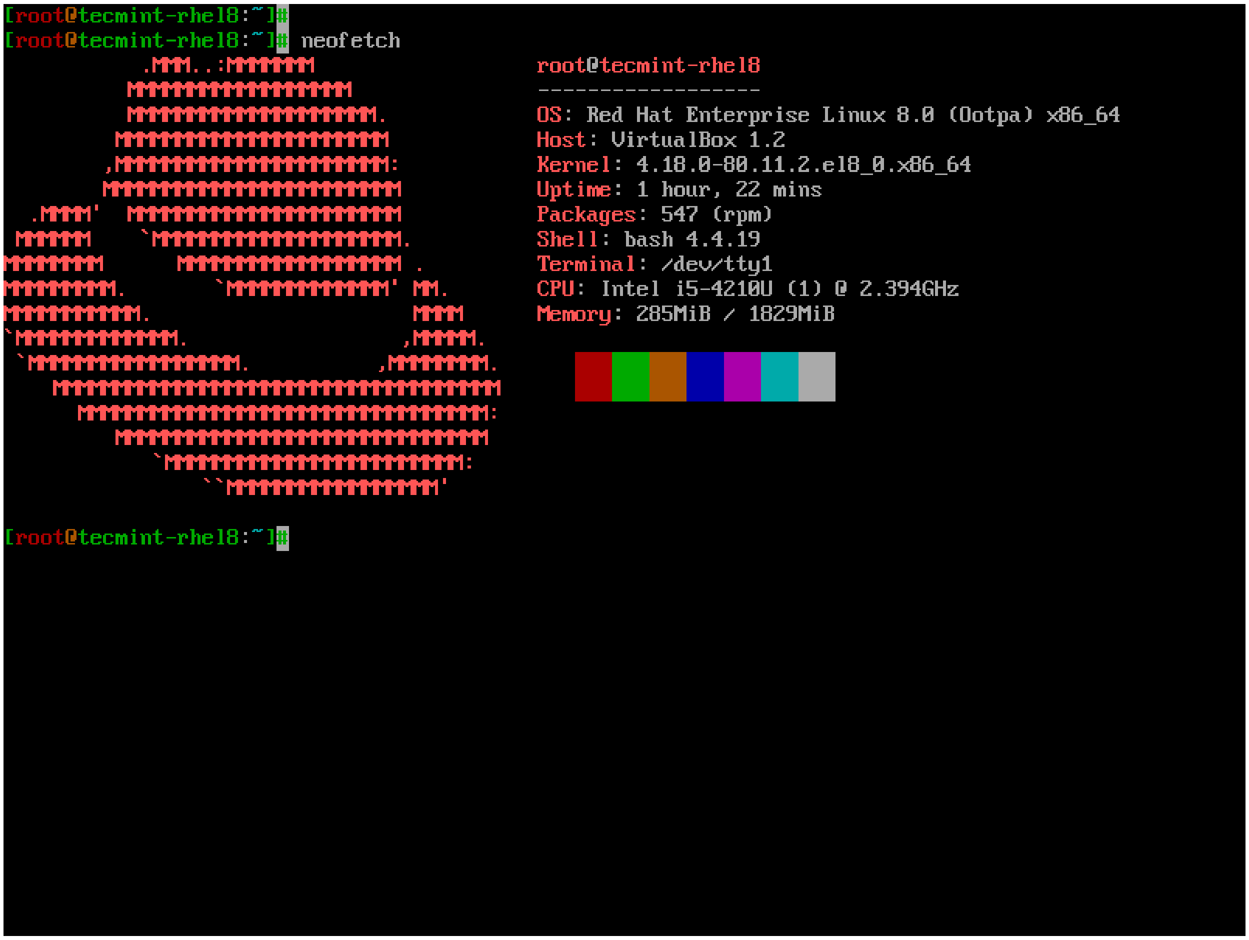
New features include **Warpinator**, which is a file-sharing program that works in a LAN and fractional scaling feature for HiDPI displays to enjoy sharper and crisp images. You will also get other applications for everyday use such as Firefox, LibreOffice, Audacious music player, Timeshift, and Thunderbird.

If you want a fast and stable Linux desktop to perform day-to-day desktop tasks, listening to music, watching videos, and even gaming, Mint is the go-to distribution. **Mint 20** is a long term release and will receive support until 2025. We have an article on how to install Mint 20 on your PC.

**Download Linux Mint ISO Image** – https://linuxmint.com/download.php

## 5. Red Hat Enterprise Linux

Abbreviated as **RHEL**, Red Hat Enterprise Linux is a Linux distro designed for Enterprise or commercial purposes. It’s one of the leading open-source alternatives to other proprietary systems such as **Microsoft**. **Red Hat** is usually a top choice for server environments given its stability and regular security patches which boost its overall security.



```
[root@tecmint-rhel8:~]#
[root@tecmint-rhel8:~]# neofetch
           .MMM..:MMMMMMM                  root@tecmint-rhel8
          MMMMMMMMMMMMMMMMMM               ------------------
          MMMMMMMMMMMMMMMMMMMM.            OS: Red Hat Enterprise Linux 8.0 (Ootpa) x86_64
         MMMMMMMMMMMMMMMMMMMMMM            Host: VirtualBox 1.2
        ,MMMMMMMMMMMMMMMMMMMMMM:           Kernel: 4.18.0-80.11.2.el8_0.x86_64
        MMMMMMMMMMMMMMMMMMMMMMMM           Uptime: 1 hour, 22 mins
  .MMMM'  MMMMMMMMMMMMMMMMMMMMMM           Packages: 547 (rpm)
 MMMMMM    `MMMMMMMMMMMMMMMMMMMM.          Shell: bash 4.4.19
MMMMMMMM      MMMMMMMMMMMMMMMMMM .         Terminal: /dev/tty1
MMMMMMMMM.       `MMMMMMMMMMMMM' MM.       CPU: Intel i5-4210U (1) @ 2.394GHz
MMMMMMMMMMM.                     MMMM      Memory: 285MiB / 1829MiB
`MMMMMMMMMMMMM.                 ,MMMMM.
 `MMMMMMMMMMMMMMMMM.          ,MMMMMMMM.
    MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
      MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM:
         MMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
            `MMMMMMMMMMMMMMMMMMMMMMMM:
                ``MMMMMMMMMMMMMMMMM'

[root@tecmint-rhel8:~]#
```

RHEL Linux Distribution

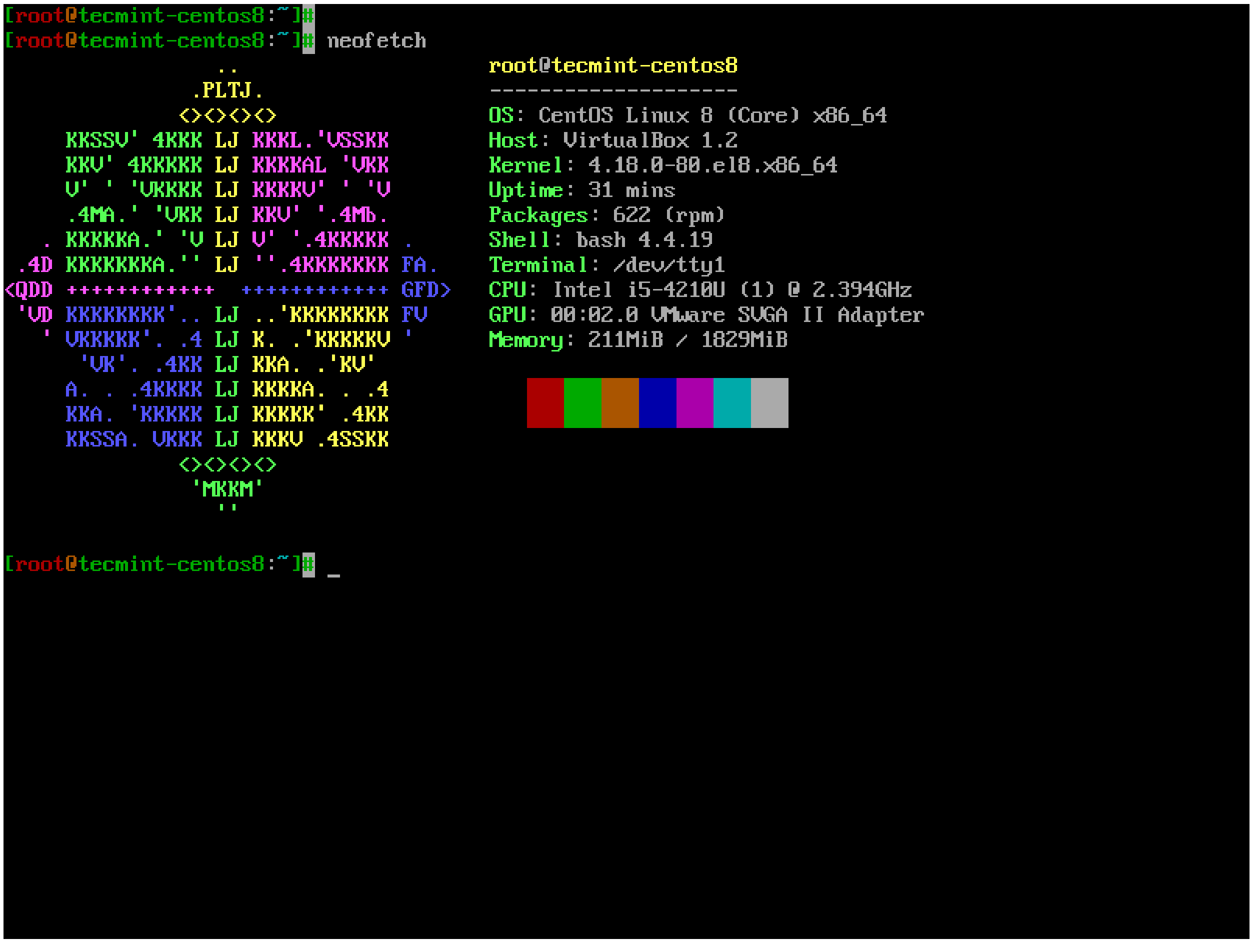
You can readily set it up on physical servers, virtual environments such as **VMware**, **HyperV,** and also on the cloud. **Red Hat** has done a perfect job in containerization technology thanks to **OpenShiftPaaS** (platform as a service), a hybrid cloud environment that is built around Docker containers and managed by Kubernetes. Redhat trains and certifies system administrators through specialist courses such as RHCSA (Red Hat Certified System Administrators) and RHCE (Red Hat Certified Engineer). Where efficiency, security, and stability are of utmost priority **RHEL** is the ideal distro to opt for. **RHEL** is subscription-based and the subscription is renewed annually. You can purchase a license for an array of subscription models such as Linux Developer Workstation, Linux developer suite, and Linux for Virtual Datacenters. Traditionally, Red Hat and its derivatives such as **CentOS** have used yum package manager. Things have changed now and the latest release – **RHEL 8 (Ootpa)** – now uses DNF as its default package manager. RHEL is distributed using 2 main repositories – **AppStream** repository and the **BaseOS**.

The **AppStream** repository (**Application Stream**) provides all the software applications that you want to install on your system while the **BaseOS** provides applications only for the core functionality of the system.

.

## 6. CentOS

The CentOS Project is a community-driven free operating system that aims at delivering a robust and reliable open source ecosystem. Based on **RHEL**, **CentOS** is a perfect alternative to **Red Hat Enterprise Linux** since it is free to download and install. It gives users the stability and reliability of **RHEL** while allowing them to enjoy free security and feature updates. **CentOS 8** is a favourite among Linux enthusiasts who want to savour the benefits of RHEL.



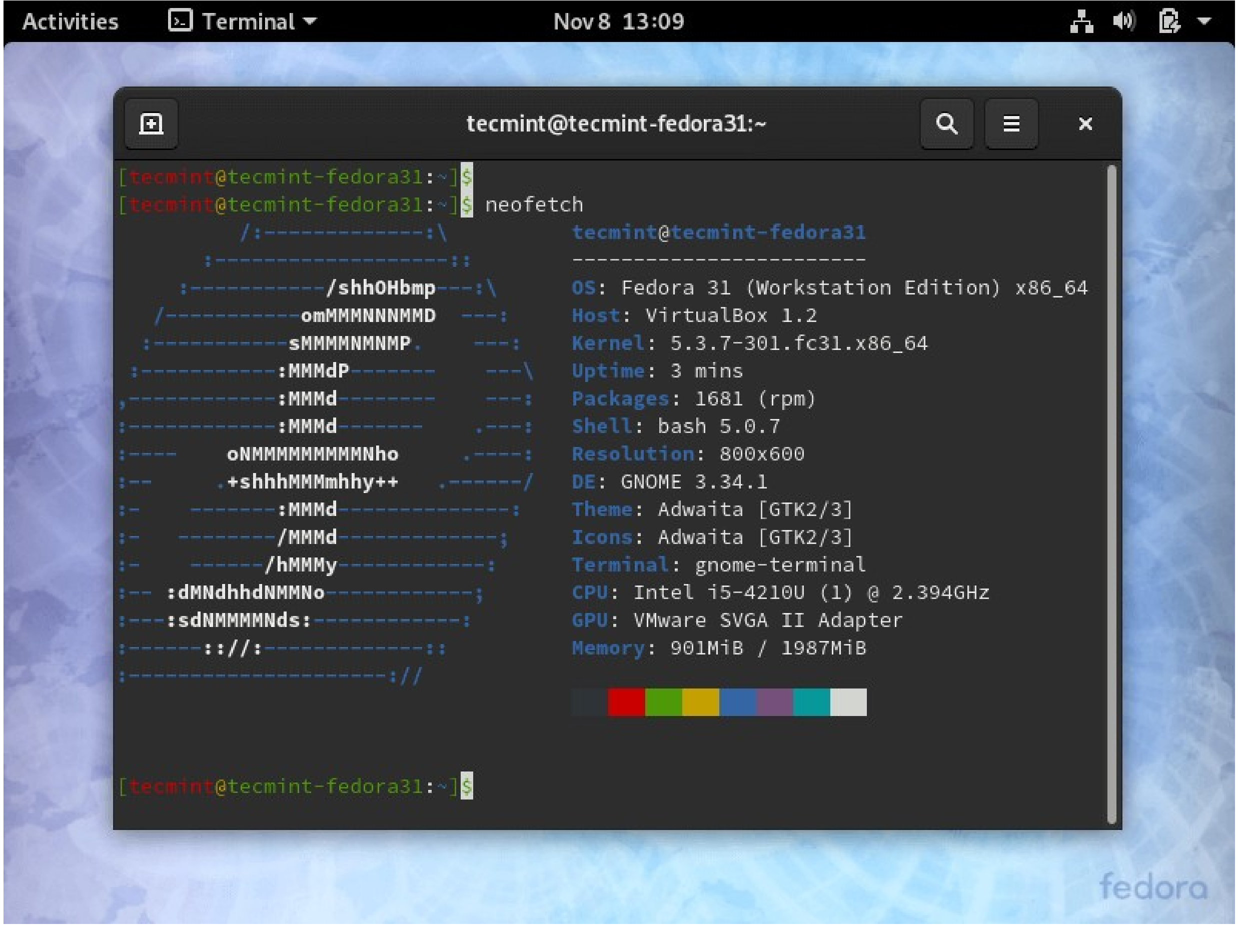
```
[root@tecmint-centos8:~]#
[root@tecmint-centos8:~]# neofetch
                 ..                    root@tecmint-centos8
               .PLTJ.                  --------------------
              <><><><>                 OS: CentOS Linux 8 (Core) x86_64
     KKSSV' 4KKK LJ KKKL.'VSSKK        Host: VirtualBox 1.2
     KKV' 4KKKKK LJ KKKKAL 'VKK        Kernel: 4.18.0-80.el8.x86_64
     V' ' 'VKKKK LJ KKKKV' ' 'V        Uptime: 31 mins
     .4MA.' 'VKK LJ KKV' '.4Mb.        Packages: 622 (rpm)
   . KKKKKA.' 'V LJ V' '.4KKKKK .      Shell: bash 4.4.19
 .4D KKKKKKKA.'' LJ ''.4KKKKKKK FA.    Terminal: /dev/tty1
<QDD ++++++++++++  ++++++++++++ GFD>   CPU: Intel i5-4210U (1) @ 2.394GHz
 'VD KKKKKKKK'.. LJ ..'KKKKKKKK FV     GPU: 00:02.0 VMware SVGA II Adapter
   ' VKKKKK'. .4 LJ K. .'KKKKKV '      Memory: 211MiB / 1829MiB
      'VK'. .4KK LJ KKA. .'KV'
     A. . .4KKKK LJ KKKKA. . .4           🟥🟩🟫🟦🟪🟦⬜
     KKA. 'KKKKK LJ KKKKK' .4KK
     KKSSA. VKKK LJ KKKV .4SSKK
              <><><><>
               'MKKM'
                 ''

[root@tecmint-centos8:~]# _
```

CentOS Linux Distribution

The latest version is **CentOS 8.2** which is the third iteration of **CentOS 8**. It relies on **App stream** and **BaseOS** repositories and ships with the latest software packages such as Python 3.8, GCC 9.1, Maven 3.6, etc.

## 7. Fedora

Fedora has enjoyed a reputation for being one of the most user-friendly distros for quite a while now owing to its simplicity and out-of-the-box applications which enable newcomers to easily get started. It's a powerful and flexible operating system that's tailored for desktops & laptops, servers, and even for IoT ecosystems. **Fedora**, just like **CentOS**, is based on **Red Hat** and is in fact, a testing environment for **Red Hat** before transitioning to the Enterprise phase. As such, it's usually used for development and learning purposes and comes in handy for developers and students.



```
[tecmint@tecmint-fedora31:~]$
[tecmint@tecmint-fedora31:~]$ neofetch
          /:-------------:\          tecmint@tecmint-fedora31
       :-------------------::        ------------------------
     :-----------/shhOHbmp---:\      OS: Fedora 31 (Workstation Edition) x86_64
   /-----------omMMMNNNMMD  ---:     Host: VirtualBox 1.2
  :-----------sMMMMNMNMP.    ---:    Kernel: 5.3.7-301.fc31.x86_64
 :-----------:MMMdP-------    ---\   Uptime: 3 mins
,------------:MMMd--------    ---:   Packages: 1681 (rpm)
:------------:MMMd-------    .---:   Shell: bash 5.0.7
:----    oNMMMMMMMMMNho     .----:   Resolution: 800x600
:--     .+shhhMMMmhhy++   .------/   DE: GNOME 3.34.1
:-    -------:MMMd--------------:    Theme: Adwaita [GTK2/3]
:-   --------/MMMd-------------;     Icons: Adwaita [GTK2/3]
:-    ------/hMMMy------------:      Terminal: gnome-terminal
:-- :dMNdhhdNMMNo------------;       CPU: Intel i5-4210U (1) @ 2.394GHz
:---:sdNMMMMNds:------------:        GPU: VMware SVGA II Adapter
:------:://:-------------::          Memory: 901MiB / 1987MiB
:---------------------://



[tecmint@tecmint-fedora31:~]$
```

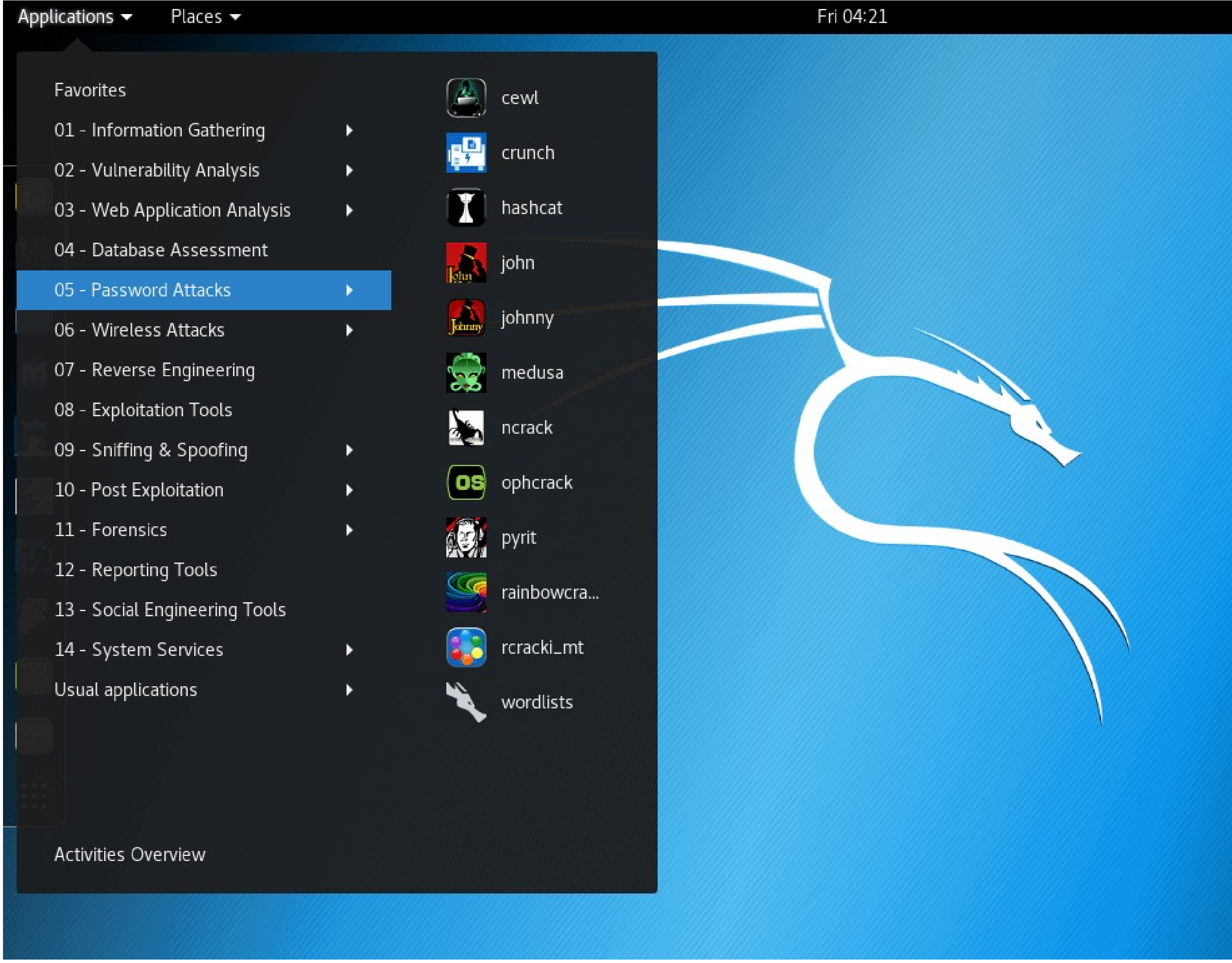
F

edora Linux Distribution

**Fedora** has for a while now used the DNF package manager ( and still uses it as its default package manager) and offers the latest and the very best in RPM software packages. The latest Fedora is **Fedora 32**.

# 8. Kali Linux

Developed and maintained by offensive security, Kali Linux is a Debian-based Linux distro designed for penetration testing and conducting digital forensics. It ships with out-of-the-box tools meant for penetration testing such as Nmap, Metasploit Framework, Maltego, and Aircrack-ng to mention a few.



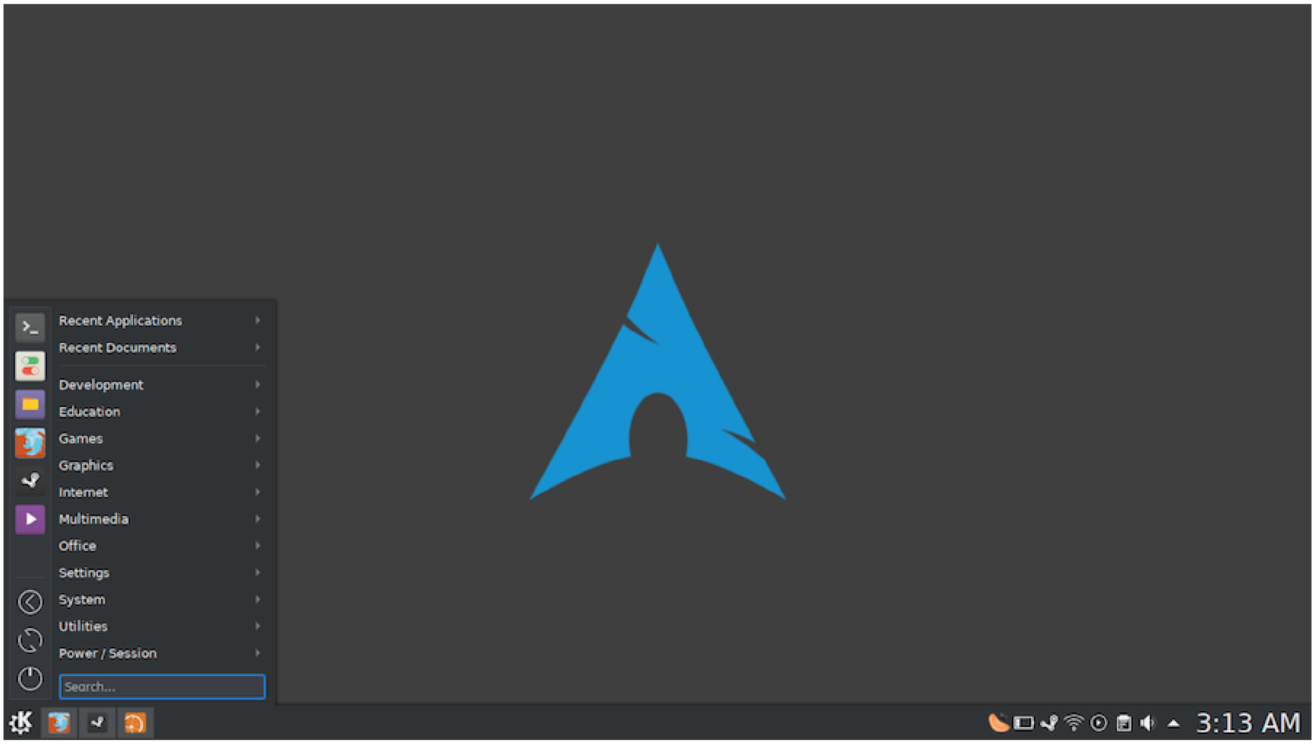
Kali Linux Distribution

**Kali Linux** is meant for Cybersecurity experts and students who want to venture into penetration testing. In fact, Kali provides industry-standard certifications such as Penetration Testing with Kali and Kali Linux Certified Professional.

**Kali** uses the **APT** package manager and the latest version is **Kali 2020.2**

## 9. Arch Linux

Arch Linux is a lightweight and flexible geeky Linux distro designed for advanced users or Linux experts who care much about what is installed and the services running. It gives users the freedom to custom or configures the system, to their preference. In a nutshell, Arch is meant for users who really know the ins and outs of working with Linux.

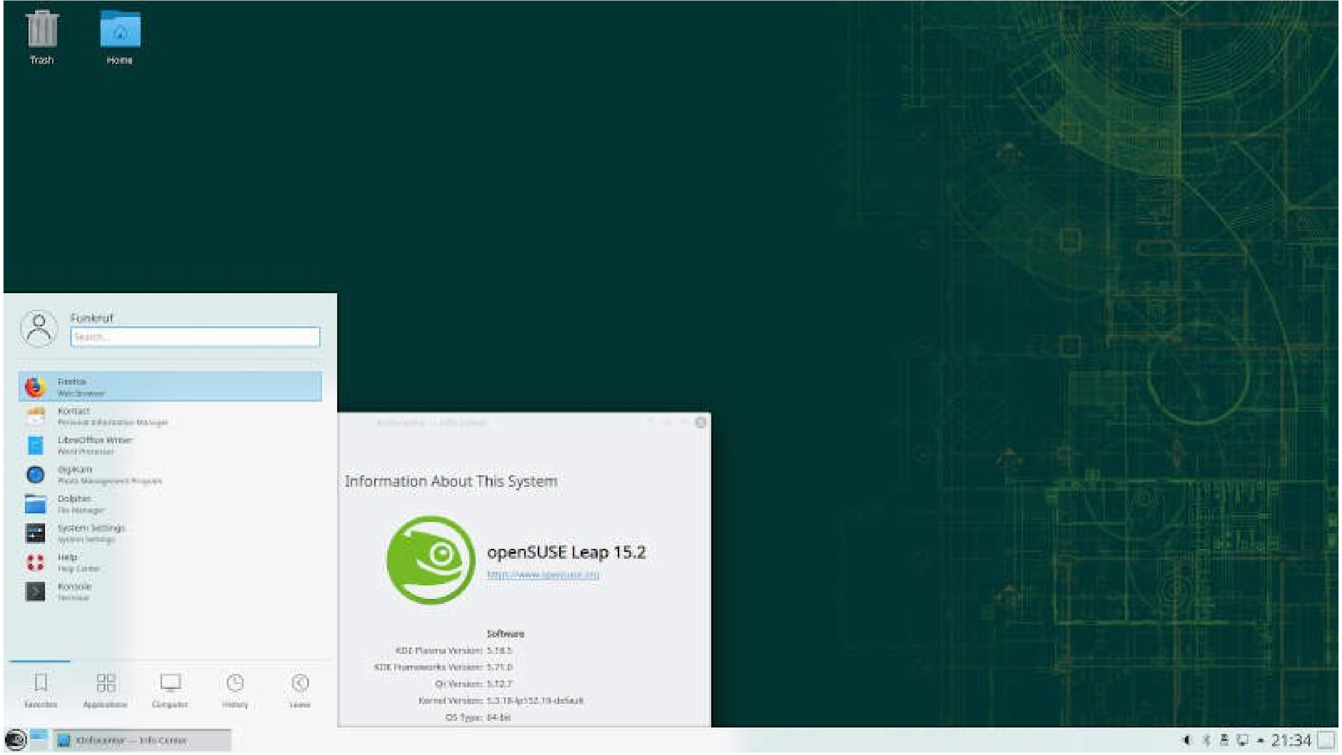


Arch Linux

**Arch** is a rolling release implying that it is constantly updated to the latest version and all you need is to update the packages on the terminal. It uses **Pacman** as the default package manager and leverages the **AUR** (**Arch User Repository**) which is a community to install software packages and the latest version is **2020.09.01**.

## 10. OpenSUSE

The OpenSUSE project is a modern and comprehensive community project that provides 2 main SUSE branches: SUSE Leap which is a point release that targets desktop users as well as enterprise development and for testing purposes. This makes it a perfect choice for open source developers and System administrators.



Information About This System

openSUSE Leap 15.2

http://www.opensuse.org

Software

KDE Plasma Version: 5.18.5
KDE Frameworks Version: 5.71.0
Qt Version: 5.12.7
Kernel Version: 5.3.18-lp152.19-default
OS Type: 64-bit

OpenSuse Linux

On the other hand, it has SUSE Tumbleweed, a rolling release that packs the latest software stacks and IDEs and is the closest you'll get to a bleeding-edge distro. **TumbleWeed** is any power user's or software developer's piece of cake thanks to the availability of up-to-date packages such as office applications, GCC compiler, and the kernel.

OpenSUSE relies on the **Yast** package manager for managing software packages and is recommended for developers and system administrators.

## Conclusion

Of course, that's just a handful of the available Linux distributions out there and is by no means an exhaustive list. There are over **600** Linux distros and about **500** in active development. However, we felt the need to focus on some of the widely used distros some of which have inspired other Linux flavors.

## Key features of Linux :-

1. Open source
2. Hierarchical File Structure
3. Packages extension - .deb & .rpm

Popular distros that use DEB:

- Ubuntu
- Debian
- Linux Mint
- Trisquel
- gNewSense

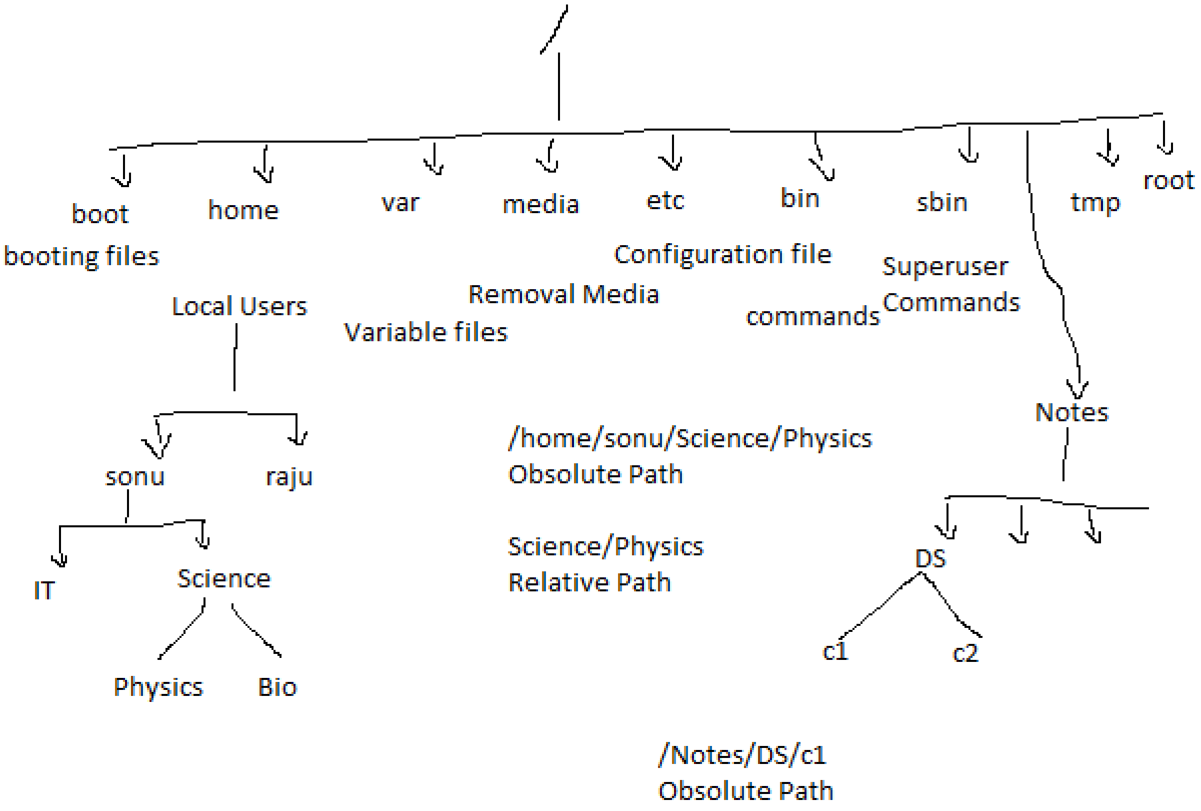
Popular distros that use RPM:

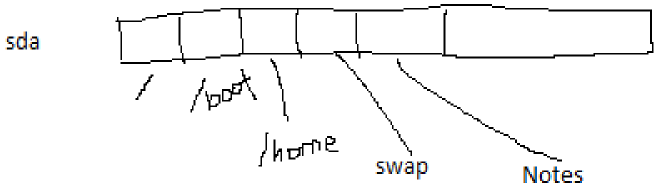
- Red Hat Enterprise Linux
- Fedora
- openSUSE
- SUSE Linux Enterprise Server
- CentOS
- PCLinuxOS

4. HDD – sd – sda, sdb, sdc
   Hd – had, hdb, hdc
   Sda – sda1, sda2,sda3
5. Administrator Account – root
6. Linux treats everything as a file.
7. It is case sensitive – Tom tomtOm, toM
8. Multiuser, Multitasking, Real Time OS
9. Virus free
10. Terminal – GUI (1)+ CLI (5) -> Function Key is used to open different terminals. F1,2,3,4,5,6,7
    F1 / F7 – GUI
    F2-F6 – CLI
11. Remote Accessing –SSH, Remote Desktop – TightVNC

12. Logical Volume Management (LVM) – Huge storage management capacity
13. Shell – provides command line tool - Korne, Borne, C Shell, Bash shell(sh)

## Linux Hierarchical File Structure :-





sda -> /  sda1
/boot - sda2
/home - sda3
swap - sda4
/home/Notes - sda5

## Now, Install Linux :-

1. Time Zone – Asia/ Kolkata
2. Language – US English
3. Keyboard – US Keyboard
4. Local User -

### What is Shell?

A shell is a program that acts as an interface between a user and the kernel. It allows a user to give commands to the kernel and receive responses from it. Through a shell, we can execute programs and utilities on the kernel. Hence, at its core, a shell is a program used to execute other programs on our system.

Most Commonly used Shell:-

### 1. The Bourne Shell (sh)

Developed at AT&T Bell Labs by Steve Bourne, the Bourne shell is regarded as the first UNIX shell ever. It is denoted as sh. It gained popularity due to its compact nature and high speeds of operation.

This is what made it the default shell for Solaris OS. It is also used as the default shell for all Solaris system administration scripts.

**However, the Bourne shell has some major drawbacks.**

- It doesn't have in-built functionality to handle logical and arithmetic operations.
- It cannot recall previously used commands.

The complete path-name for the Bourne shell is /bin/sh and /sbin/sh. By default, it uses the prompt *#* for the root user and *$* for the non-root users.

### 2. The GNU Bourne-Again Shell (bash)

More popularly known as the Bash shell, the GNU Bourne-Again shell was designed to be compatible with the Bourne shell. It incorporates useful features from different types of shells in Linux such as Korn shell and C shell.

It allows us to automatically recall previously used commands and edit them with help of arrow keys, unlike the Bourne shell.

The complete path-name for the GNU Bourne-Again shell is /bin/bash. By default, it uses the prompt *bash-VersionNumber#* for the root user and *bash-VersionNumber$* for the non-root users.

### 3. The C Shell (csh)

The C shell was created at the University of California by Bill Joy. It is denoted as csh. It was developed to include useful programming features like in-built support for arithmetic operations and a syntax similar to the C programming language.

Further, it incorporated command history which was missing in different types of shells in Linux like the Bourne shell. Another prominent feature of a C shell is “aliases”.

The complete path-name for the C shell is /bin/csh. By default, it uses the prompt *hostname#* for the root user and *hostname%* for the non-root users.

### 4. The Korn Shell (ksh)

The Korn shell was developed at AT&T Bell Labs by David Korn, to improve the Bourne shell. It is denoted as ksh. The Korn shell is essentially a superset of the Bourne shell.

Besides supporting everything that would be supported by the Bourne shell, it provides users with new functionalities. It allows in-built support for arithmetic operations while offereing interactive features which are similar to the C shell.

The Korn shell runs scripts made for the Bourne shell, while offering string, array and function manipulation similar to the C programming language. It also supports scripts which were written for the C shell. Further, it is faster than most different types of shells in Linux, including the C shell.

The complete path-name for the Korn shell is /bin/ksh. By default, it uses the prompt *#* for the root user and *$* for the non-root users.

### 5. The Z Shell (zsh)

The Z Shell or zsh is a sh shell extension with tons of improvements for customization. If you want a modern shell that has all the features a much more, the zsh shell is what you’re looking for.

**Some noteworthy features of the z shell include:**

- Generate filenames based on given conditions
- Plugins and theming support
- Index of built-in functions
- Command completion
- and many more…

Let us summarise the different shells in Linux which we discussed in this tutorial in the table below.

| **Shell** | **Complete path-name** | **Prompt for root user** | **Prompt for non root user** |
|---|---|---|---|
| Bourne shell (sh) | /bin/sh and /sbin/sh | # | $ |
| GNU Bourne-Again shell (bash) | /bin/bash | bash-VersionNumber # | bash-VersionNumber$ |
| C shell (csh) | /bin/csh | # | % |
| Korn shell (ksh) | /bin/ksh | # | $ |
| Z Shell (zsh) | /bin/zsh | <hostname># | <hostname>% |

## Few important features of Bash Shell : –

1. # -> root user / Super-user / Administrator
2. $ -> local user

3.  Single Tab – it will complete the command

4. Double Tab – it will give all options

   exit
   enable
   enter

5. Switch User

   su<username>   -> New user will login with current user profile

   su - <username>  -> New user will login with its own profile

6. Promot - >     [root@localhost~]#

   User, computer name, ~ home directory of user

7. Remote Accessing : ssh ( Secure Shell )
8. Stores history of commands :  #history

   #!command serial no.

## Commands :-

1. pwd  - present working directory
2. ls – list directory and files
3. ls –l – list directory or files with detail information

## File Handling Commands :

4. touch<filename>    -> used to create blank file.
5. touch .file-name     -> To create hidden file

6. mv<old file name><new file name >    -> used to rename a file.
7. mv<old file ><same file with new loacation>    -> to move a file at different location.
8. cp<source file><destination>      -> to copy a file
9. rm<file name >                -> To delete a file
10. rm –rf<file name>           -> To delete a file forcefully
11. cat ><file-name>        -> To create a new file and take input
12. cat >><file-name>     -> To append into existing file
13. cat ><existing file name >   -> Over write
14. cat <file-name>         -> To see the content of file
15. vi <file-name>                                        ----→ Text editor in CLI

    Different modes of vi – a> Command Mode
    b> Insert Mode
    c> Execute Mode

    vim<file-name>

    i -> To go into insert mode

    Press Esc ->To come out from insert mode

    :wq          -> To save and quit

    :wq!        -> To save and quit forcefully

    Notes :- Text Editor in GUI -→gedit, kwrite
16. #wc<file name>  -> To count word, line and characters

    #wc –l <To show line>

    #wc –w < To show word>

    #wc –c  < To show characters>
17. #find < file name>    -> To search a file into current dir
18. #locate <file name>  -> To search a file into entire system

    Notes : To run locate command, database must be updated

    #dbupdate

    ******* Blue line is applicable for RHEL by default ****
19. #head <file name>

#head -4 <file name >  -> To show top specified lines

20. #tail <file name>

    #tail -6 <file name>   ->To show bottom specified lines.

21. File compression – To reduce to size of file

    #gzip<file name>     ->To compress a file

    #gunzip<compressed file name>-> To decompress a file

22. #du –b/-k/-m <file name> -> To check the size of a file / dir

23. Backup of file and compression :

    #tar –cvzf file-name

24. Restore and decompression of file

    #tar –xvzf file-name

25. #grep root /etc/passwd

    #grep‘root[0-9]’ /etc/passwd

    #grep‘root[^0-9]’ /etc/passwd

26. #cut –d: -f1,3 /etc/passwd> report

27. #sort filename > sort1                    -> Used to sort a file

    #sort filename > sort2

28. #comm sort1 sort2                    -> Used to find common

29. #diff sort1 sort2                    -> Used to find difference

30. #uniq sort1 sort2                    -> Used to find unique

31. #tr ‘a-z’ ‘A-Z’  < file-name                    -> Used to change case of letter

32. To see the file page by page

    #cat /etc/passwd | more

33. To take input from a file

    #wc –l  < /etc/passwd

Directory Handling Commands :-

1. #mkdir<dir name><dir name>
2. #cd /home/sonu/songs -> To change dir
3. #cd .. -> One stop back dir
4. #cd -> To go into home dir
5. #cp –rf<source><destination> -> To copy dir
6. #mv <source><destination-same location> -> To rename a dir
7. #mv <source><destination – at diff location> -> To move a dir
8. #rmdir<dir name> -> Delete empty dir
9. #rm –rf<dir with contents> -> To delete dir with content

Few more commands :

1. #sleep 10s/m/h/d
2. #expr 25 + 10
3. #dd if=/dev/sda of=/dev/sdb
4. #sed ‘s/cow/dog/’ file-name
5. #factor 10 20 30

## AWK command in Unix/Linux with examples

Awk is a scripting language used for manipulating data and generating reports. The awk command programming language requires no compiling and allows the user to use variables, numeric functions, string functions, and logical operators.

Awk is a utility that enables a programmer to write tiny but effective programs in the form of statements that define text patterns that are to be searched for in each line of a document and the action that is to be taken when a match is found within a line. Awk is mostly used for pattern scanning and processing. It searches one or more files to see if they contain lines that matches with the specified patterns and then perform the associated actions.

Awk is abbreviated from the names of the developers – Aho, Weinberger, and Kernighan.

**WHAT CAN WE DO WITH AWK?**

**1. AWK Operations:**

(a) Scans a file line by line
(b) Splits each input line into fields
(c) Compares input line/fields to pattern
(d) Performs action(s) on matched lines

### 2. Useful For:

(a) Transform data files

(b) Produce formatted reports

### 3. Programming Constructs:

(a) Format output lines
(b) Arithmetic and string operations
(c) Conditionals and loops

### Syntax:

```
awk options 'selection _criteria {action }' input-file > output-file
```

**Options:**

```
-f program-file : Reads the AWK program source from the file
                  program-file, instead of from the
                  first command line argument.
-F fs           : Use fs for the input field separator
```

### Sample Commands

### Example:

Consider the following text file as the input file for all cases below:

```
$cat > employee.txt
ajay manager account 45000
sunil clerk account 25000
varun manager sales 50000
amit manager account 47000
tarun peon sales 15000
deepak clerk sales 23000
sunil peon sales 13000
satvik director purchase 80000
```

**1. Default behavior of Awk:** By default Awk prints every line of data from the specified file.

```
$ awk '{print}' employee.txt
```

**Output:**

```
ajay manager account 45000
sunil clerk account 25000
varun manager sales 50000
amit manager account 47000
tarun peon sales 15000
deepak clerk sales 23000
sunil peon sales 13000
satvik director purchase 80000
```

In the above example, no pattern is given. So the actions are applicable to all the lines. Action print without any argument prints the whole line by default, so it prints all the lines of the file without failure.

**2. Print the lines which match the given pattern.**

```
$ awk '/manager/ {print}' employee.txt
```

**Output:**

```
ajay manager account 45000

varun manager sales 50000

amit manager account 47000
```

In the above example, the awk command prints all the line which matches with the ‘manager’.

**3. Splitting a Line Into Fields :** For each record i.e line, the awk command splits the record delimited by whitespace character by default and stores it in the $n variables. If the line has 4 words, it will be stored in $1, $2, $3 and $4 respectively. Also, $0 represents the whole line.

```
$ awk '{print $1,$4}' employee.txt
```

**Output:**

```
ajay 45000

sunil 25000

varun 50000

amit 47000

tarun 15000

deepak 23000

sunil 13000

satvik 80000
```

In the above example, $1 and $4 represents Name and Salary fields respectively.

**Built-In Variables In Awk**

Awk’s built-in variables include the field variables—$1, $2, $3, and so on ($0 is the entire line) — that break a line of text into individual words or pieces called fields.

- **NR:** NR command keeps a current count of the number of input records. Remember that records are usually lines. Awk command performs the pattern/action statements once for each record in a file.
- **NF:** NF command keeps a count of the number of fields within the current input record.
- **FS:** FS command contains the field separator character which is used to divide fields on the input line. The default is “white space”, meaning space and tab characters. FS can be reassigned to another character (typically in BEGIN) to change the field separator.
- **RS:** RS command stores the current record separator character. Since, by default, an input line is the input record, the default record separator character is a newline.
- **OFS:** OFS command stores the output field separator, which separates the fields when Awk prints them. The default is a blank space. Whenever print has several parameters separated with commas, it will print the value of OFS in between each parameter.
- **ORS:** ORS command stores the output record separator, which separates the output lines when Awk prints them. The default is a newline character. print automatically outputs the contents of ORS at the end of whatever it is given to print.

**Examples:**

**Use of NR built-in variables (Display Line Number)**

```
$ awk '{print NR,$0}' employee.txt
```

**Output:**

```
1 ajay manager account 45000
2 sunil clerk account 25000
3 varun manager sales 50000
```

```
4 amit manager account 47000
5 tarun peon sales 15000
6 deepak clerk sales 23000
7 sunil peon sales 13000
8 satvik director purchase 80000
```

In the above example, the awk command with NR prints all the lines along with the line number.

**Use of NF built-in variables (Display Last Field)**

```
$ awk '{print $1,$NF}' employee.txt
```

**Output:**

```
ajay 45000
sunil 25000
varun 50000
amit 47000
tarun 15000
deepak 23000
sunil 13000
satvik 80000
```

In the above example $1 represents Name and $NF represents Salary. We can get the Salary using $NF , where $NF represents last field.

**Another use of NR built-in variables (Display Line From 3 to 6)**

```
$ awk 'NR==3, NR==6 {print NR,$0}' employee.txt
```

**Output:**

```
3 varun manager sales 50000
4 amit manager account 47000
5 tarun peon sales 15000
6 deepak clerk sales 23000
```

# Vim Editor in Linux

**Vim** is an advanced and highly configurable text editor built to enable efficient text editing. Vim text editor is developed by Bram Moolenaar. It supports most file types and vim editor is also known as a programmer's editor. We can use with its plugin based on our needs

## Installation and Configure vim in Our Linux System

To install vim on Debian based Linux like ubuntu run the command:

```
sudo apt-get install vim
```

To install vim on an arch-based distro run the following command:

```
sudo pacman -S vim
```

Now vim will be installed on your system.

There are some commands given to use the vim editor. You can see all commands, and it's documentation by help command as follows:

```
:help
```

Now to exit type command

```
:!q
```

## Now, Let's Start to use Vim

To open a file in vim editor just write the file name after the vim command in the terminal as follows:

```
vim filename.txt
```

Then the file will be opened.

### Write into file

In the previous step we have opened the file now, Let's write some content in to write data we need to go in insert mode. To go into write mode type i.  As follows:

```
i
```

After going into insert mode you will see INSERT in the status bar. After that, we can write any data in it.

**Save and Exit:**
We have written the data into a file now the task is to save and close the file to do that first exit from insert mode by pressing the Esc key. To write a command first type semicolon `( : )` and then type the command wq  And then hit ENTER.

```
:wq
```

**Exit without saving the file:**
To exit from the file without saving the file just use the command q As follows

```
:q
```

Vim also comes with its own tutorial. You can see this tutorial by command vimtutor into the terminal .

```
vimtutor
```

**Moving the Cursor:**

Till now, we are using the arrow keys to move the cursor into a file, but it is not recommended to use arrow keys in vim. Vim provides the special key to move the cursor in the vim editor following are the keys used to move into a file.

```
* k   ->   move up
* j   ->  move down
* h   -> move right
* l   -> move left
```

Just use these commands in vim files and move the faster cursor into the files.

**Exiting Vim:**
To exit without doing anything go into command mode by pressing the Esc key and type the following command.

```
:q
```

To exit from vim without saving changes type the following command in vim

```
:q!
```

To quit and save changes type the following command

```
:wq
```

**Text Editing: Deletion**
We provide x key in command mode to delete the character under the cursor. Move the cursor to the character which has to delete and press Esc key  and then press the x key

x

The character under the cursor will be deleted.

**Text Editing: Insertion**
We have edited some text files before by using the i key. There are Four keys used for insertion text. Just type the key into the normal mode in vim.

```
 i    ->   This key is used to put the
cursor before the current position.

 a    ->   This key is used to put the
cursor after the current position.

 o    ->   This key is used to put the
cursor below the line.
```

```
 O    ->   This key is used to puts the cursor above the line.
```

**Motion:** Motions provide context to your Operators. These execute the action in a particular way.
Here is a list of some motions

```
w - until the start of the next word, EXCLUDING its first character.
```

```
e - to the end of the current word, INCLUDING the last character.
```

```
$ - to the end of the line, INCLUDING the last character.
```

We can use motion with the d key and with many more keys

**Count:** Count is the number for which replete the motion for count number. Here is a demonstration of the use of count and motion
To move courser 2 words forward use the following command

```
2w
```

Here 2 is the number of counts and w is used for word

To move cursor 4 line forward use the following command

```
4$
```

**Deletion Commands:** Always use the Esc key to go into normal mode and use the insertion, deletion keys, and other keys.
To **delete the word** move the cursor to the beginning of the word and use dw command in normal mode. The word under the cursor will be deleted.

```
dw
```

To delete more than one word in a single line use the following command.

To delete 2 words use the command

```
d2w
```

To **delete the line** move cursor to the beginning of the line and use d$ command in normal mode. The line under the cursor will be deleted.

```
d$
```

**Undo and Redo:**
As we are programmers most time we are using undo and redo .vim to provide these to both features in it. To undo press u key in normal mode

```
u
```

To redo use the ctrl+r key in normal mode in vim

```
ctrl+r
```

**Replace :**
To replace the character under the cursor use rx command where ‘x’ is a character is to replace.

```
r
```

**Change Operator:**
In vim c is used as a change operator. To replace the word use ce command

```
ce
```

To replace the line use c$ command

```
c$
```

This command will delete the contains from the cursor to the end of the line. Then this will automatically go into insert mode then you can put anything in that line,

**Cursor location:**
We can use the h,j,k,l key to move the cursor in the file but it will be difficult to move in big file vim provide more commands to move into the file.

To move the cursor at the start of the file use the gg command

```
:gg
```

To move the cursor at the bottom of the file use the G command

```
:G
```

To view the current cursor location in the file using the following command:

```
:ctrl+g
```

In place of 32 use can put any line number.

**Search:**
To search the word After the cursor uses the backslash key and then write the word and press enter.

```
:/word
```

Use n to move on next matching word

```
:n
```

Use N to move on previous matching word

```
:N
```

**Search and Replace:**
To replace the word in file use s/ command in vim like

```
:s/searchword/replaceword/
```

To do replace all occurrence of word use g

```
:s/searchword/replaceword/g
```

This command will replace the word globally.

To confirm before replacements of words use gc

```
:s/searchword/replaceword/gc
```

To use this command in the whole file use % before the command

```
:%s/searchword/replaceword/gc
```

**Vim configuration:**
To configure vim, vim comes with the .vimrc file in the home directory if this file is not there then create one. Then open the file in vim by command

```
vim ~/.vimrc
```

You can put your all configurations in this file.

You can set line number by using a simple command

```
set number
```

Put this command in your .vimrc file

To enable syntax highlighter in vim use command

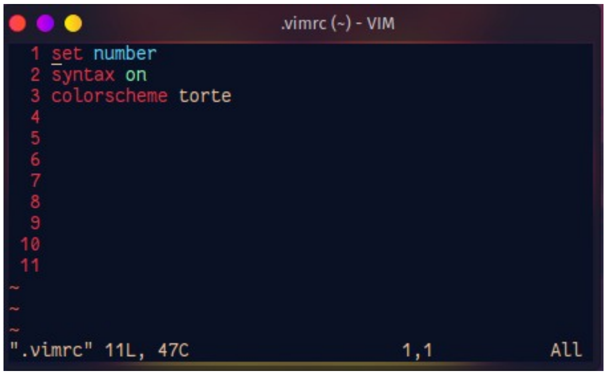
```
syntax on
```

Put this command in .vimrc file then save file and exit after opening file again you will see the line numbers in vim.

Vim comes with many color schemes using the command:

```
colorscheme COLOR_SCHEME_NAME
```

Replace COLOR_SCHEME_NAME with any color scheme between Default, blue, darkblue, delek, desert, elford, evening, industry, koehler, morning, murphy, pablo, peachpuff, ron, shine, slate, torte, zellner then save file and exit then again open file to see changes.



```
set number
syntax on
colorscheme torte








~
~
~
".vimrc" 11L, 47C                   1,1            All
```

## User Account Handling :-

1. Linux User – Super User – root
   Local User / Standard User – user created by admin.
2. Users are identified by name and an ID.
3. Numeric Range of User ID -   0 – 999999
4. Root user ID – 0
5. Reserved User ID -  0 – 999  - reserved for system generated user at the time of installation of Linux OS.
6. User ID for manually created user – starts from 1000
7. User Configuration file
   #cat /etc/passwd | more
   0 0 x root root /root /bin/bash
   0 – userid
   0 – groupid
   X – password link
   Root – username
   Root – group name to which user belongs.
   /root – user home directory

/bin/bash – login shell

Notes : whenever any user is created , a group is also created by OS.

8. All group information file :
   #cat /etc/group | more
9. Password information of users : #cat /etc/shadow | more
10. Passwords are in encrypted mode with MD5(Message Digest 5)
11. How to create user :
    # useradd<username>
    Or,
    # adduser<username>
12. How to define password :
    # passwd<username>
13. How to create group :
    # groupadd<groupname>
14. How to add a user into a group :-
    # usermod –G <username><groupname>
    Example :
    User- raju
    Grup – boys
    #usermod –G raju boys
15. Modification with user : -
    To lock user account :
    # usermod –L <username>
    To unlock a user :
    # usermod –U <username>

To change login name :

```
# usermod –l <new login name ><old login name>
```

To delete a user account :

```
#userdel<username>   - user account will be deleted without home directory.
# userdel –r <username> - user account will be deleted with home directory
```

Link

Link of a file is created for easy access and to stop accidental deletion of file.

Types of Link –

1. Hard Link
2. Soft Link / Symbolic Link

Hard Link –

a. It is created in the same drive / partition.

b. It uses same Inode value

   Inode Value – It is used to represent a file. It is numeric value. Its range depends upon the size of partition.

   We can see inode value of a file by –

   ```
   # ls –i <file name >
   ```

c. It does not represent the original file.

d. If link file is deleted , no effect on original file and if original file is deleted , no effect on link file.

e. Edit in any will reflect into all files.

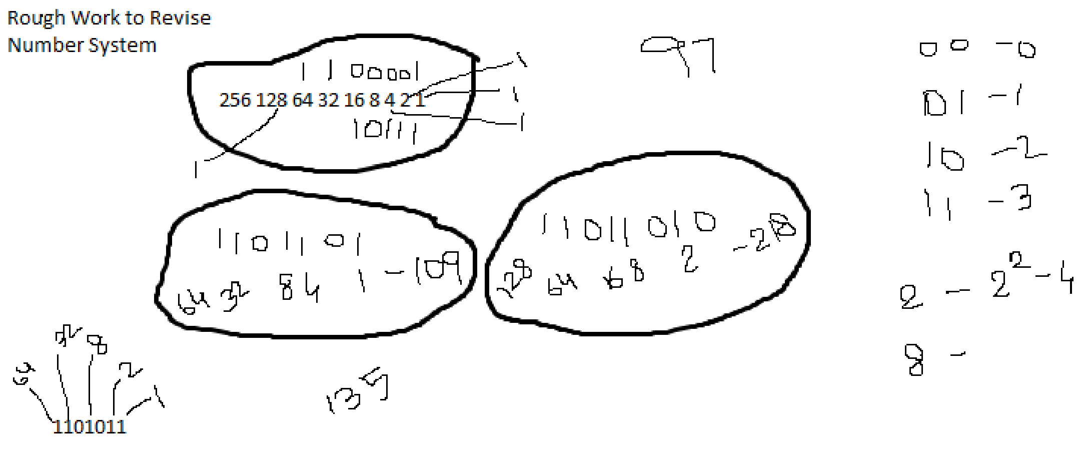
f. How to created hard link of file?

#ln<original file name><link file name>

Soft Link –

a. It can be created in the same drive or different drive.
b. It uses different inode value for each link file.
c. It reprents the original file with pointer.
d. If original file is deleted, link file cannot be accessed.
e. #ln –s <original file> <link file name>

Networking :-


Rough Work to Revise Number System

## Address

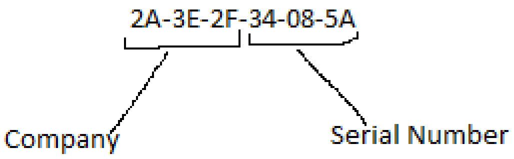
Address represents identity of node, components or services in IT infrastructure. It is required for successful transmission of data.

Different forms of Address :

1. Physical Address
2. Logical Address
3. Port Address
4. Special Address

## Physical Address

1. This address is available at all those devices which can be connected to network through wired or wireless media.
2. Physical address is present at NIC ROM.
3. It is also known as Hardware Address / MAC address.
4. It is attached with data frames at Data Link Layer of OSI Ref. Model. So, It is helpful in node to node delivery of data frames.
5. Ex. 2A-3D-4F-00-74-8C
6. Broadcast MAC Address = FF-FF-FF-FF-FF-FF
7. It is represented into Hexadecimal.
8. It has 12 Hexadecimal characters ( 48 Bits )
9. First 6 characters represents manufacturer company and last 6 characters represents serial number given by manufacturer company. So, this address is always globally unique.
10. Commands to see MAC address :
    a. C:>getmac
    b. C:>ipconfig /all
    c. ifconfig



## Logical Address

1. This address is managed by Internet Protocol (IP). So, it is also known as IP address.
2. It works at Network Layer of OSI Ref. Model.
3. It is a combination of number that can be managed manually or through DHCP server. So, it is known as logical address.
4. It provides unique identity to network nodes locally or globally.
5. It is responsible for source to destination delivery of data packets.

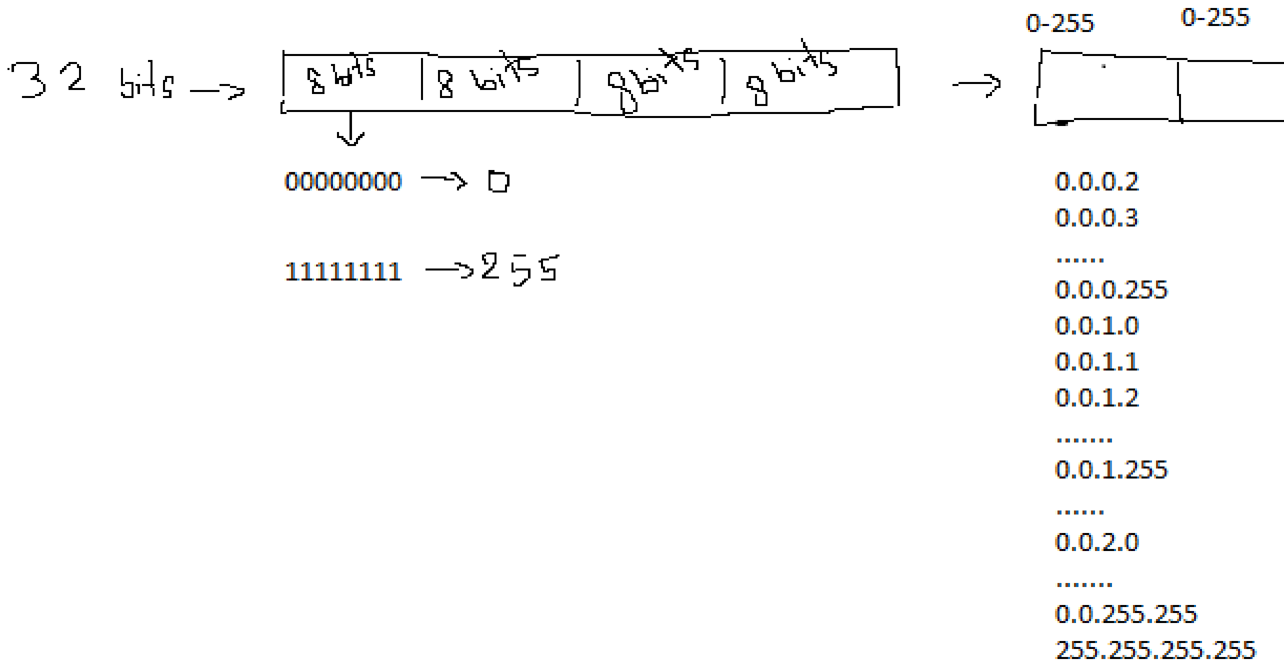
Types of IP address / Version of IP address :

1. IPv4 Address
2. IPv6 Address

### IPv4 Address :

1. Developed by DARPA ( Defence Advanced Research Projects Agency ) in 1981.
2. Managed and distributed by IANA ( Internet Assigned Number Authority ).
3. Represented into dotted decimal number.
4. It has 32 bits divided into 4 groups each having 8 bits.
5. It supports approx 3.2 million IPv4 addresses.
6. It supports unicasting, multicasting and broadcasting transmission way.
7. It is divided into 5 classes. -> A, B, C,D,E

Now, we will understand the concept of IPv4 address:



| | | | |
|---|---|---|---|
| 0-255 | | | |

The value of first octet is divided to form different classes.

0-127 -> Class A

128-191 -> Class B

192-223 -> Class C

224-239 -> Class D

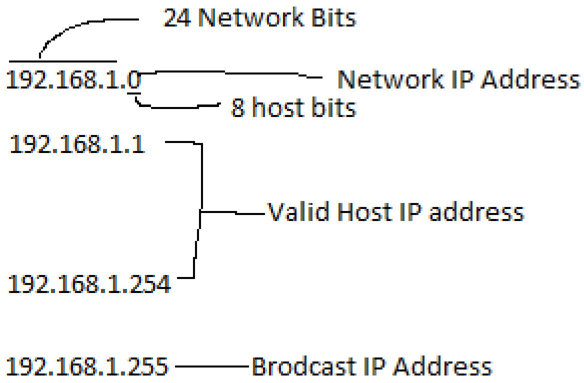
240 -255 -> Class E

Few important concepts before understanding these classes:

1. Network Bits – This value defines different series of IP address.
2. Host Bits – This value defines number of hosts in a network series.
3. Network IP address – The very first IP address of a network series which defines the a network.
4. Broadcast IP address – The very last IP address of a network series which is used for broadcasting in the network

5. Subnet-mask – This value defines the range of IP address in a network and defines different networks.

Example:

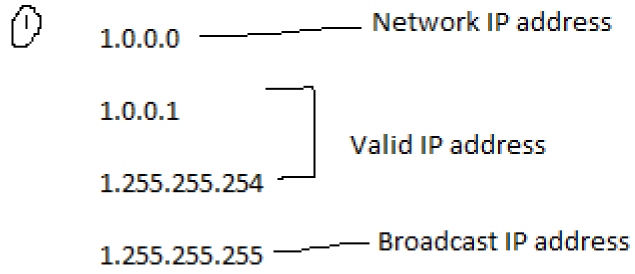


## Class A

| | | | |
|---|---|---|---|
| 0-127 | 0-255 | 0-255 | 0-255 |
| Net. Bits | Host Bits | Host Bits | Host Bits |

Important Points:

1. It has 8 bits of network and 24 bits of host.
2. Number of default networks = 128
3. Number of valid hosts IP address in each default network = 16777216-2=16777214
4. Default subnetmask = 255.0.0.0
   Example:

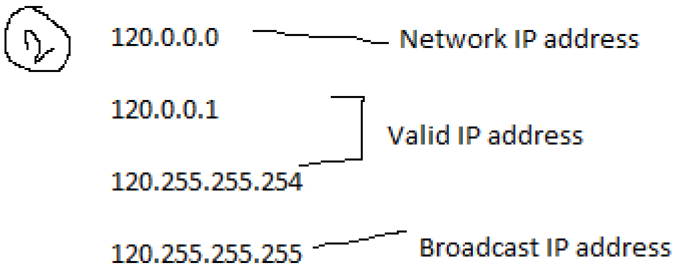


1

1.0.0.0 — Network IP address

1.0.0.1 – 1.255.255.254 — Valid IP address

1.255.255.255 — Broadcast IP address



2

120.0.0.0 — Network IP address

120.0.0.1 – 120.255.255.254 — Valid IP address

120.255.255.255 — Broadcast IP address

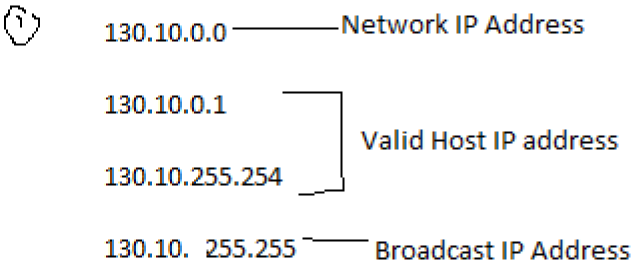
## Class B

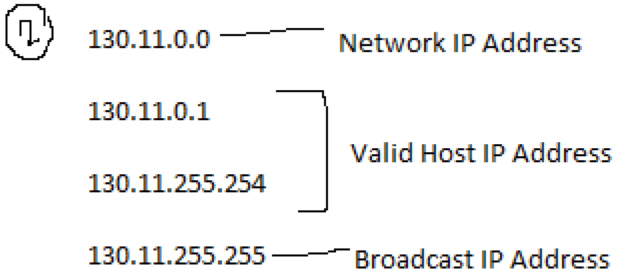
| | | | |
|---|---|---|---|
| 128-191 | 0-255 | 0-255 | 0-255 |
| Net. Bits | Net. Bits | Host Bits | Host Bits |

Important Points:

1. It has 16 bits of network and 16 bits of host.
2. Number of default networks = 16384
3. Number of valid hosts IP address in each default network = 65536-2=65534
4. Default subnetmask = 255.255.0.0

   Example:





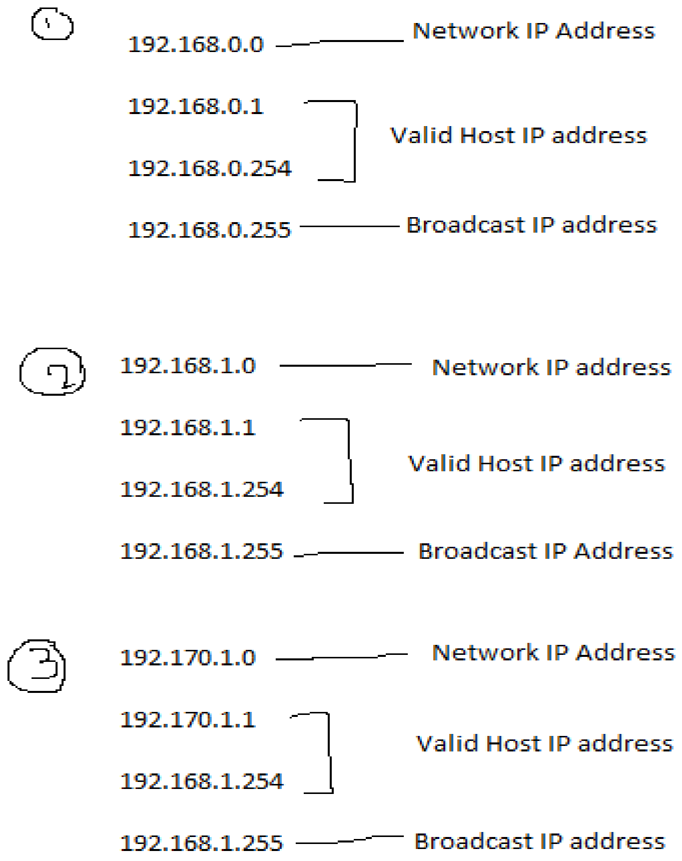
## Class C

| | | | |
|---|---|---|---|
| 192-223 | 0-255 | 0-255 | 0-255 |
| Net. Bits | Net.Bits | Net. Bits | Host Bits |

Important Points:

1. It has 24 bits of network and 8 bits of host.
2. Number of default networks = 2097152
3. Number of valid hosts IP address in each default network = 256-2=254
4. Default subnetmask = 255.255.255.0

Example:



## Class D

Few Points :

1. All 32 bits represents network.
2. No any host bits
3. It is used for multicasting only by different services( Protocol ), in TV channels, etc.
   Example:
   OSPF uses 224.0.0.5 and 224.0.0.6
   RIP – 224.0.0.9
   EIGRP – 224.0.0.10
   NTP – 224.0.1.1
   DHCP – 224.0.0.12
   IGMP – 224.0.0.22

mDNS – 224.0.0.251
Link Local Multicast IP Address – 224.0.0.252

## Class E

This class is not used for any purpose.

## Reserved IP Addresses:

Class A -> 0.0.0.0 -> Reserved for Default route

127.0.0.1 -> Reserved for loopback testing

Class B -> 169.254.x.y -> Reserved for APIPA

## Private IP Address

These addresses are used in local network. It does not require any reservation before use. In Class A, B and C few series have been defined as Private IP address.

Class A -> 10.0.0.0 to 10.255.255.255

Class B -> 172.16.0.0 to 172.31.255.255

Class C -> 192.168.0.0 to 192.168.255.255

## Public IP Address

**After** excluding reserved and private IPv4 addresses, rest IP addresses of class A,B and C are Public IP addresses. It is used for global unique identity in the network.

Network Connection :

```
$ nmcli

$ nmcli device status

$ nmcli connection show

$ nmcli connection show --active

$ nmcli -f DEVICE, TYPE device

$ nmcli -p device

$ sudo nmcli con add type ethernet con-name "static-ip" ifname enp0s3
ipv4.addresses 192.168.2.150/24 gw 192.168.2.1

$ sudo nmcli con mod static-ip ipv4.dns "8.8.8.8 8.8.4.4"

$ sudo nmcli con up static-ip ifname enp0s3

$ sudo nmcli con down id "static-ip" ifname enp0s3

$ sudo nmcli con down id "static-ip"

$ sudo nmcli con up id "static-ip" ifname enp0s3
```

IPv4 configuration at Ubuntu machine :

Vi /etc/netplan/01-..........

```
ethernets:
           enpos3:
               dhcp4: no
addresses: [10.0.0.2/8]
                gateway4: 10.0.0.1
```

```
nameservers: [10.0.0.1,8.8.8.8]
:wq
#netplan apply
#ping 10.0.0.2
```

**Linux Permission :**

```
# ls –l     -> file permission
drw_rw_r__
rw-      -> Owner permission
rw-     -> Group permission
r--       ->Other user permission
```

Accessor of file / dir ->

Owner – u

Group – g

Other User – o

Permissions Representation :-

Read -   r  -  4

Write -  w  -  2

Execute – x  -  1

All permission - a - 7

Read write - rw- -> 6

Read execute - r-x -> 5

No permission - Numerical– 0 -rwx

Implementation of permission :-

*Alphabetical Method :*

+ Add permission to existing permission

- Remove permission from existing permission

= Remove all existing permission and add these permission only.

Implementation :

To give read, write and execute permission to owner(u)

u+rwx (4+2+1=7)

g-wx (4)

o=wr (6)

746

u+rwx,g-wx,o=wr

Permission Command :-

```
#chmod  u+rwx,g-wx,o=rw <file / directory name>

#chmod u+rwx <file/directory name>

#chmod –r/p u+rwx,g-wx,o=rw <file / directory name>

#ls –l

#chmod 746 <file/directory name>
```

Notes :-> File or directory permission depends upon its parent directory or drive.

## Access Control List (ACL) :-

Example :

```
#chmod 660 corona

#chmod 600 covid
```

ACL is used to provide specific permission to specific user or group at a file of directory.

File – apple

One user from other user – raja

#setfacl –m u:raja:rw apple  -> it is used to apply special permission at file/dir.

#getfacl apple        -> it is used to see the special permission at file/dir.

#setfacl –x u:raja: apple -> it used to remove special permission from a file /dir.

How to give special permission to a specific group :

1. Create a group
   #groupadd boys
2. Create some user
   #useraddashish
   #useraddrahul
   #passwdashish
   123
   #passwdrahul
   123
3. Add created user into the group :
   #usermod –G ashish boys
   #usermod –G rahul boys
   #vim /etc/groups
4. Give special permission to a group with ACL
   #setfacl –m g:boys:rw apple
   #getfacl apple
   #setfacl –x g:boys: apple

## Umask :-

It is default numeric value that is always subtracted from the default permission of file / directory.

#umask

0022

```
$umask
0002
```

Default permission of a file or directory

File : 666

Dir  : 777

When root creates a file/directory:

File -> 666 – 022 = 644

Dir ->  777 – 022 = 755

Example : File ->rw-r—r--

When local user creates file / directory

File - > 666 – 002 = 664

Dir  -> 777 – 002 = 775

Example : File ->rw-rw-r—

Umask can be used to set default permission at a file or directory.

```
#umask 077
```

File -> 666 – 077 = 600

Dir ->  777 – 077 = 700

Example : File ->rw--—---

Dir  ->rwx------

To change umask :

#umask 022

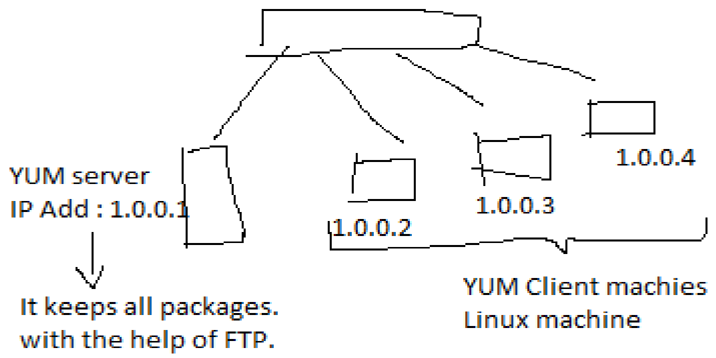
Special Permission :

1. Sticky Bit(1) – This permission is used for files into a directory from stop deletion by other users.
   # mkdir river
   #cd river
   #touch ganga Yamuna ravi satlaj
   #cd
   #chmod 1777 river
2. Sgid (2) – This permission is used to fix the group owner of files stored into a directory
   # mkdir city
   User – ronak
   Touch Delhi
   Owner – ronak
   Group owner - ronak
   # chmod 2777 city
   User – ronak
   Touch ranchi
   Owner – ronak
   Group Owner – Saqlain
3. Suid (4) – This permission is used to give privilege to local user to use commands that can be used by superuser only.
   #chmod 4777 /usr/sbin/useradd
   Ronak$ useradd tiger

Linux Package Management :

1. Linux packages are used to install different services and applications.
2. Linux packages are available with .rpm extention.
3. RPM ( Redhat Package Manager )
4. Packages are available online and with linuxos image.
5. To install a package :
   #rpm –ivh<package name>
6. To see installed package :
   #rpm –q <package name>
7. To remove a installed package
   #rpm –e <package name>

   Notes : Few packages shows dependency on another packages
   Dependency problem is solved with YUM server.
8. YUM ( Yellowdog Update Modified )-
   This service provides us facility to install linux packages with dependeny and through network also.
   All packages are kept at central location( FTP ).
   Packages from this location can be access through command line and web browser.



Steps : At server machine ->

1. Configure an ip address at server machine.
2. Go into package directory
   #cd /run/root/RHEL....../Appdata/Packages
   #rpm –ivhvsftpd-------
   #systemctl start vsftpd.service
   #systemctl status vsftpd.service
   Notes : It will create a directory /var/ftp/pub/
   #cp –avf /run/root/RHEL..../Appdata/Packages /var/ftp/pub/
   Now, check pub data
   #ls –l /var/ftp/pub/
   Now, we have to create repository into /var/ftp/pub/Packages for indexing of collected packages. It creates a file with extention .xml

```
#rpm –ivh
/run/root/RHEL..../Appdata/Packages/createrepo......
#createrepo /var/ftp/pub/Packages
```

Steps : For both server and client machines

```
#setenforce 0

#systemctl stop firewalld.service

#systemctl disable firewalld.service

#systemctl status firewalld.service
```

Write a script to create repo file :

```
#vim /etc/yum.repoes.d/sonam.repo
```

Go into insert mode :

```
[sonam]
name=sonam
baseurl=ftp://1.0.0.1/pub/Packages
gpgcheck=0
```

Now, save the above file.

```
#setenforce 0
```

Used to disable linux security

```
#systemctl stop firewalld.services
#systemctl disable firewalld.service
#systemctl status firewalld.service
```

Now, our YUM server is ready.

How to check it.

```
#yum repolist
```

How to install a package with yum :

```
#yum install tightvnc
#yum install vsftpd
```

How to remove a installed packages :

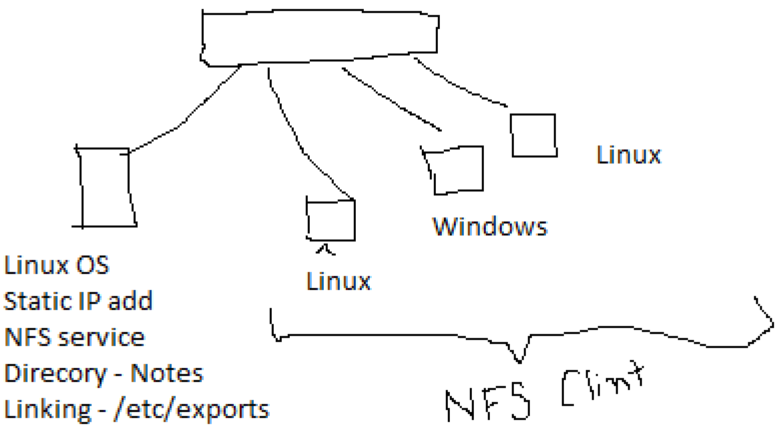
```
#yum remove tightvnc
```

How to update a package :

```
#yum update <package name>
```

## Network File System ( NFS )

It is used to share data in a network through Linux.
It is supported by Windows OS also.
A Linux computer at which NFS is configured is known as NFS server.
The latest version of NFS is NFS4.
A shared directory is linked with /etc/exports file.
After sharing it can be accessed through network with temporary or permanent mounting.



Steps :

For NFS server :

Install NFS package

```
#rpm –ivh nfs4---------
#mkdir /notes
#touch /notes/windows
#touch /notes/linux
#chmod 777 /notes
#vim /etc/exports
/notes    *(sync,rw)
:wq
#setenforce 0
```

```
# systemctl stop firewalld.service
#systemctl disable firewalld.service
#systemctl status firewalld.service
#systemctl start nfs.service     Or, #systemctl start nfs-server.service
#showmount –e <nfs server ip address>
/notes *
```

Configuration at client :linux

```
#setenforce 0
# systemctl stop firewalld.service
#systemctl disable firewalld.service
#systemctl status firewalld.service
#mkdir /nfsdata
#showmount –e <nfs server ip add>
```

Temporary Mounting :-

```
#mount <nfs server ip add>:/notes    /nfsdata
#ls –l /nfsdata
```

Permanent Mounting :

```
#vim /etc/fstab
```

Go to last line of file –

```
<nfs server ip address>:/notes /nfsdata nfs  defaults 0 0
```

How to access NFS data at windows machine :

Control Pannel -> Programs and features -> Turn windows features on or off -> select NFS service -> ok -> install

Open My Computer -> Map network drive -> path \\nfs server ip add\notes ->dirive letter Z -> ok

Now, you will get all data of NFS server into a separate drive.

CIFS ( Common Internet File System ):

Share a directory(Folder) at windows machine with full permission.

SAMBA :-

This service is used to configure SMB protocol at Linux machine. With the help of SMB, we can share date for Windows machine.

How to configure Samba service at Linux server ?

```
#rpm –ivh samba-4---------
```

```
#rpm –ivh samba-client------
```

```
#rpm –ivh samba-common-------
```

Or,

```
#yum install samba*
```

Now, create a directory with full permission :-

```
#mkdir /city
```

```
#chmod 777 /city
```

Keep some data into directory:

```
#touch /city/Mumbai
```

```
#touch /city/delhi
```

Now, configure smb.conf file for Samba configuration :

```
#vim /etc/samba/smb.conf
```

Go to bottom of file:

```
[ajeet]
```

```
path = /city

writable = yes

browseable = yes

users = ajeet
```

Now, save the file.

```
#setenforce 0
```

Disable firewall.

Create a user and make it a part of samba

```
#useraddajeet

#passwdajeet

123

#smbpasswd –aajeet

Password :123

123

#service smb restart
```

Now, go at windows PC.

Run ->\\IP address of linux machine

Username :ajeet

Password : 123

FTP ( File Transfer Protocol )

It is used to transfer file through command line and browser in the network. Port No. 21

How to configure FTP server?

```
#rpm –ivhvsftpd---------
```

It will create a location -> /var/ftp/pub/

Create a directory into pub

```
#mkdir /var/ftp/pub/jetdata
```

Keep some file into jetdata

```
#touch /var/ftp/pub/jetdata/f1
#touch /var/ftp/pub/jetdata/f2
```

Now, start the vsftpd service

```
#service vsftpd restart
```

Stop Linux security

```
#setenforce 0
```

Disable firewall service

Now, ftp can be accessed through default user Anonymous.

Make some changes with ftp script..

```
#vim /etc/vsftpd/vsftpd.conf
```

```
#anonymous_upload=yes    -> remove hash from this line to provide upload permission.
```

Now, go to client.

Linux ->

Install ftp package for command.

```
#rpm –ivh ftp-------
```

```
#ftp <ftp server ip add>
```

```
User: anonymous

Password : press enter key

ftp>cd pub/jetdata

ftp>mget f1     -> to download file from ftp server

ftp>mputajeet  -> to upload local file at ftp server

ftp>bye
```

Windows :-

```
C:\>ftp <ftp server ip add>

Username : anonymous

Password :

ftp>get pub/jetdata/f1

ftp>put pub/jetdata/ajeet
```

To block anonymous user :

Go to ftp server

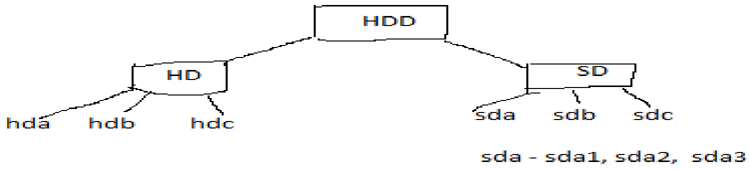
```
#vim /etc/vsftpd/vsftpd.conf
```

Anonymous_enabl=YES   -> Change this YES into NO

## Storage Management :-

HDD – Storage Media

In case linux –



Partitions :

Primary – Max. 4 No. Extended Partition

Primary – Max. 3     1 extended partition -> 11 logical drives = Total 15

```
#fdisk –l /dev/sda
```

```
#du /dev/sdaor  #fd /dev/sda
```

How to create partition :

```
#fdisk /dev/sda
```

Press p to list partition

Press n to create new partition

Give space -> +5G

Press p to see the newly created partition

Sda6

Press w to save created partition.

Now, give information to kernel for HDD changes.

```
#partprobe /dev/sda
```

Now, format the newly created partition.

```
#mkfs.xfs –L software /dev/sda6
```

Now, mount the formatted partition into a directory.

```
#mkdir /software
```

```
#mount /dev/sda6 /software
```

Notes : But this mounting is known as temporary mounting.

Permanent Mounting :

```
#vim /etc/fstab
```

```
/dev/sda6        /software        xfs        defaults
       0 0
```

How to check mounting?

```
#mount –a
```

```
#mount
```

To unmount a mounted partition :

```
#umount /dev/sda6
```

## LVM(Logical Volume Management)

It is used to manage multiple partitions at a disk or multiple disks.

Steps:

1. Connect few HDD with your machine.
2. Install lvm package at machine.
   ```
   #cd /run/media/root/RHEL………./Packages
   #rpm –ivh lvm2-2…………
   ```
3. Create a physical volume
   ```
   #pvcreate /dev/sd{b,c,d}
   ```
4. Create a volume group
   ```
   #vgcreate <give any name> /dev/sd{b,c,d}
   ```
5. Create logical volume
   ```
   #lvcreate –n <give name to partition> -L 10G <give volume group name>
   ```
6. To display-
   Physical volume – pvdisplay
   Volume Group – vgdisplay
   Logical Volume – lvdisplay
7. Format logical volume
   ```
   #mkfs.ext4 /dev/group name/volume name
   ```

8. Mount formatted volume

   ```
   #mkdir /songs
   #mount /dev/pat/songs /songs
   #mount –a
   #mount
   ```